(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 July 2006 (27.07.2006)

PCT

(10) International Publication Number
WO 2006/078302 A1

(51) International Patent Classification:
G06F 17/50 (2006.01)

(21) International Application Number:
PCT/US2005/020369

(22) International Filing Date: 8 June 2005 (08.06.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/644,244 14 January 2005 (14.01.2005) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 60/644,244 (CIP)
Filed on 14 January 2005 (14.01.2005)

(71) Applicant (*for all designated States except US*): **THE REGENTS OF THE UNIVERSITY OF CALIFORNIA** [US/US]; 1111 Franklin Street, 5th Floor, Oakland, CA 94607-5200 (US).

(72) Inventors; and
(75) Inventors/Applicants (*for US only*): **CHENG, Chung-kuan** [US/US]; 4407 Mensha Place, San Diego, CA 92130 (US). **ZHU, Zhengyong** [US/US]; 9152 F Regents Road, La Jolla, CA 92037 (US).

(74) Agent: **AI, Bing**; Fish & Richardson P.C., 12390 El Camino Real, San Diego, CA 92130 (US).

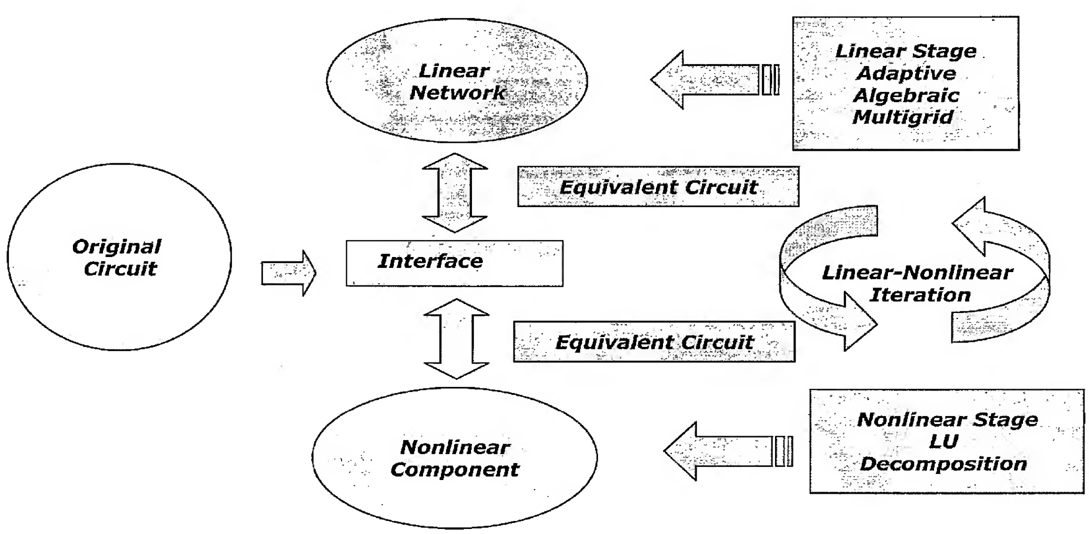
(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US (patent), UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: EFFICIENT TRANSISTOR-LEVEL CIRCUIT SIMULATION



(57) Abstract: Techniques for performing circuits with nonlinear circuit components such as transistors based on a two-stage Newton-Raphson approach.

WO 2006/078302 A1

EFFICIENT TRANSISTOR-LEVEL CIRCUIT SIMULATION

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 60,644,244 entitled "EFFICIENT TRANSISTOR LEVEL SIMULATION USING TWO-STAGE NEWTON-RAPHSON AND MULTIGRID METHOD" and filed on January 14, 2005.

Background

[0002] This application relates to analysis of integrated circuits.

[0003] Circuits may be viewed as networks of nodes and circuit components connected between nodes. As such, circuits may be analyzed based on a nodal analysis where a nodal equation may be written for each node based on the conservation of charge at the node, i.e., the total current entering the node is equal to the total current leaving the node (the Kirchoff's second rule). For a circuit with N nodes, N equations for the N nodes can be expressed in terms of the properties of circuit components such as resistance, capacitance, and inductance, and in terms of the node voltages and currents. These N equations can be written into a matrix equation and are solved using various matrix approaches such as LU decompositions of matrices. For circuits with certain control sources, inductance and current control sources, additional equations for different current branches may be added to fully describe the circuits.

[0004] Power network analysis may be performed based on the circuit matrix equations to investigate behaviors of circuit networks such as voltage drop, voltage oscillation, and electromigration. Excessive voltage drops may reduce the switching speed as well as the noise margins of circuits and may even cause logic failures under some circumstances. Electromigration may decrease the chip lifetime. Moreover voltage oscillation may occur when power network resonance frequency drops to the range of the signal frequency.

[0005] One bottleneck of the power network analysis based on the above nodal analysis is the tremendous amount of variables in large power networks such as integrated circuits. One well-known circuit analysis software program for solving such circuit equations is the SPICE circuit simulator originally developed by the University of California at Berkeley. The SPICE uses the LU decomposition of a matrix to solve the circuit equations. As the number of the circuit elements and nodes increases, the super linear complexity of the LU deposition method and other direct methods make them prohibitive for large-scale circuits.

[0006] The power networks are mostly linear circuit networks. There are other more complex circuits such as circuits with transistors which exhibit nonlinear circuit properties. Matrices for circuits with transistors may be solved by, e.g., direct simulation methods like LU decomposition used in the Berkeley SPICE2 simulator and its variations. See, Nagal, "Spice2: A computer program to simulate semiconductor circuits," Tech. Rep. ERL M520, Electronics Research Laboratory Report, UC Berkeley (1975). The direct simulation methods may become less effective and can reach their computational limits when the number of transistors in circuit networks increases and the circuit configurations become complex. This is in part because the super linear complexity $O(n^{1.5})$ increases with the number of circuit nodes, n, and the amount of the extracted interconnect data for a large n can exceed the capacity of the software based on a direct simulation method.

[0007] Various other transistor-level simulation methods have been developed to address the limitations of the direct simulation methods, some of which, however, have trade-offs in certain aspects such as the simulation accuracy in comparison with the direct simulation methods. Examples of other transistor-level simulation methods include application of

different integration methods (explicit or implicit) on subcircuits according to their activities by Sakallah and Director in "SAMSON2:An event driven VLSI circuit simulator," IEEE Trans. On Computer-Aided Design ICs and Systems, Vol. 4(4), pp.668-684 (1985), and reduction of the number and cost of LU decompositions by using low cost integration approximation and simpler linearization via successive C=chord method) by Li and Shi in "SILCA: Fast-yet-accurate time-domain simulation of VLSI circuits with strong parasitic coupling effects, " ICCAD, pp. 793-799 (2003). Various fast simulation tools for transistor-level simulation are also commercially available from companies such as Nassda Corporation (www.nassda.com), Synopsys, Inc.(www.synopsys.com), Apache Design Solutions, Inc. (www.apachedesignsolutions.com) and Cadence Design Systems (www.cadence.com). These commercial simulation software tools usually partition a circuit into subcircuits which can be simulated with different time steps to explore latency but can have potential convergence problems due to the coupling effects or strong feedback in the circuits. The convergence rate may also be sensitive to the partition algorithm and propagation order.

Summary

[0008] This application describes, among others, transistor-level simulation techniques that are computationally efficient, provide assured convergence, and provide a high accuracy that is at least comparable to the accuracy of many direct simulation methods such as SPICE for simulating VLSI circuits with strong coupling effects (e.g., deep-submicron and nanometer VLSI circuits). In described implementations, an adaptive algebraic multigrid method for large networks of power/ground, clock and signal interconnects can be used to simulate the linear part of the circuit. Transistor devices in the circuit can be integrated using a

two-stage Newton-Raphson method to dynamically model the linear network and nonlinear devices interface.

[0009] In one implementation, a method for analyzing a circuit network having transistors is described as follows. The circuit is represented by an equivalent circuit that comprises at least one linear network and at least one nonlinear cell to interface with each other. Circuit equations for the at least one linear network and the at least one nonlinear cell, respectively, are then solved. The solutions to the at least one linear network and the at least one nonlinear cell are matched at an interface between the at least one linear network and the at least one nonlinear cell. Accordingly, an article is also described to include at least one machine-readable medium that stores machine-executable instructions for the above method.

[0010] In another implementation, a method for analyzing a circuit network having transistors is described to represent a circuit by an equivalent circuit that comprises at least one linear network and at least one nonlinear cell to interface with each other. A Norton equivalent circuit is used to represent an interface between the at least one linear network and the at least one nonlinear cell. A Newton-Raphson iteration is applied to solve circuit equations for the at least one linear network and the at least one nonlinear cell. The solutions to the at least one linear network and the at least one nonlinear cell are matched at the interface between the at least one linear network and the at least one nonlinear cell.

Accordingly, an article is also described to include at least one machine-readable medium that stores machine-executable instructions for the above method. The instructions cause the machine to represent a circuit by an equivalent circuit that comprises at least one linear network and at least one nonlinear cell to interface with each other;

use a Norton equivalent circuit to represent an interface between the at least one linear network and the at least one nonlinear cell; apply a Newton-Raphson iteration to solve circuit equations for the at least one linear network and the at least one nonlinear cell; and match solutions to the at least one linear network and the at least one nonlinear cell at the interface between the at least one linear network and the at least one nonlinear cell.

[0011] These and other implementations and variations are described in greater detail in the attached drawings, the detailed description, and the claims.

Brief Description of Drawings

[0012] FIG. 1 is a block diagram showing one implementation of the present circuit simulation based on a two-stage Newton-Raphson procedure.

[0013] FIGS. 1A, 1B, 1C, 1D and 1E show equivalent circuits and the circuit I-V responses for simulating an original circuit under the two-stage Newton-Raphson procedure.

[0014] FIGS. 2A, 2B and 2C show a specific example of the equivalent circuits based on the methods illustrated in FIGS. 1A, 1B and 1C.

[0015] FIGS. 3A, 3B, 4A and 4B show details of obtaining the equivalent Norton circuits for the nonlinear device interface and the linear network interface.

[0016] FIG. 5 illustrates one example of the processing steps in implementing the two-stage Newton-Raphson procedure.

[0017] FIGS. 6A, 6B and 6C illustrate the equivalent circuit block diagrams and corresponding circuit matrix equations for linear nodes, nonlinear nodes and interface nodes.

[0018] FIGS. 7A, 7B, 8A and 8B show test results in simulating exemplary circuits using the two-stage Newton-Raphson approach based on the C programming language.

Detailed Description

[0019] The circuit simulation techniques described here are designed to simulate circuits at the transistor level using a two-stage Newton-Raphson and multigrid method. The present techniques can be used to provide a fast, accurate transistor level simulation tool for full-chip simulation including power network and clock network analysis with SPICE-level accuracy and guaranteed convergence. This simulation tool can be used to analyze various circuit properties and behaviors, such as interconnect delay, crosstalk, voltage drop, ground bounce, and simultaneous switching. The two-stage Newton-Raphson method is designed to dynamically model the linear network and nonlinear devices boundary. A fast linear solver, such as an algebraic multigrid method, can be used to solve the coupled linear network. Mutual inductance in a circuit can be incorporated in the simulation without relying on error-prone matrix sparsification approximations or computation-intensive and time-consuming (i.e., computationally "expensive") matrix inversion.

[0020] FIG. 1 illustrates one implementation of the present circuit simulation based on a two-stage Newton-Raphson procedure. The original circuit, which may be a circuit network with transistors (e.g., a VLSI circuit), is transformed into an equivalent circuit shown on the left hand side in FIG. 1 for the purpose of simulation. The equivalent circuit is partitioned into three modules for simulation: one or more equivalent linear networks that represent the linear portion of the original circuit, one or more equivalent nonlinear circuit cells that represent the nonlinear portion of the original circuit such as transistors, and one or more linear-nonlinear interfaces between the linear networks and the nonlinear circuit cells. The linear network is treated as a regular linear network by using a linear network circuit simulator. One example is an adaptive algebraic multigrid

approach as disclosed in PCT application No. PCT/US04/17237 entitled "Circuit Network Analysis Using Algebraic Multigrid Approach" and filed on June 1, 2004, which was published as PCT Publication No. WO2004109452A3 on December 16, 2004. The entire disclosure of the PCT publication is incorporated herein by reference as part of the specification of this application. The nonlinear circuit cells are also treated by using a nonlinear LU decomposition. The interface is then solved to match the linear network and the nonlinear circuit cells via linear-nonlinear iterations as graphically represented in FIG. 1.

[0021] The above partitioning or decoupling of the nonlinear circuit devices and the linear networks of a given circuit is made inside the Newton-Raphson iteration and the linear-nonlinear interface is dynamically modeled via a two-stage Newton-Raphson procedure. The accurate modeling of the linear-nonlinear interface inside the Newton-Raphson iterations ensures the convergence of the simulation.

[0022] FIGS. 1A, 1B and 1C show graphic representation of the above simulation technique. FIG. 1A shows the equivalent circuit with a nonlinear cell and a linear network with an interface in the middle. Each linear-nonlinear interface is modeled as a Norton equivalent circuit, i.e., a circuit having a single current source characterized by an equivalent current Ieq and a load characterized by an equivalent conductance Geq. FIG. 1B shows that the nonlinear cell is treated as an equivalent Norton circuit coupled to the linear network. FIG. 1C shows that the linear network is treated as an equivalent Norton circuit coupled to the nonlinear cell. The equivalent Norton circuit for the linear network is different from the equivalent Norton circuit for the nonlinear cell and the differences are indicated by the circuit parameters: the branch current and nodal voltage at the interface, the current of the current source and the conductance of the load. In

FIGS. 1A-1C, the equivalent conductance and equivalent current parameters are Geq and Geq, Ieq and Ieq, respectively. The parameters V, V, I and I are nodal voltages and branch currents, respectively.

[0023] Consider the interface boundary condition as an I - V function i = f(z), the Newton-Raphson concept can be used to model the interface. More specifically, a discrete Newton-Raphson method is implemented as shown in FIGS. 1A-1C and includes two stages: a nonlinear stage for nonlinear transistor devices and a linear stage for linear network. The interfaces are linearized as Norton equivalent circuits. After this linearization, the linear networks and nonlinear cells can be solved separately inside each Newton-Raphson iteration.

[0024] FIG. 1D illustrates an example of the I-V function at a linear-nonlinear interface. FIG. 1E shows an equivalent piecewise linearized curve that represents the I-V function in FIG. 1D. The linearized equation from the Newton-Raphson method is:

$$i = i^k + \frac{di}{dv}\left(v - v^k\right) \qquad (1)$$

The dynamic conductance and the equivalent current source can be obtained as below:

$$\begin{cases} G_{eq} = \dfrac{di}{dv}\Big|k \\ I_{eq} = G_{eq}v^k - i^k \end{cases} \qquad (2)$$

[0025] The nonlinear devices have given nonlinear I - V functions and analytic G_{eq} formulation. The I-V function f(v) at the interface boundary is unknown. According to the linearity property and the problem size of nonlinear cell and

linear network, different discrete differential approximations are implemented for nonlinear cells and linear networks.

[0026] As an example, FIGS. 2A, 2B and 2C show the equivalent circuits in simulation of the an NAND gate. FIG. 2A shows the NAND gate circuit surrounded by coupled power, ground and signal networks. Decoupling and interface modeling of device and linear network are shown in FIGS. 2B and 2C for the equivalent nonlinear cells and linear network, respectively. Each connection with or to the NAND gate is simulated by a Norton equivalent circuit in FIGS. 2B and 2C. The transistor bulk nodes can also considered in the simulation.

[0027] Referring back to FIGS. 1A-1B, the details of obtaining the equivalent Norton circuits for the nonlinear device interface and the linear network interface are described below using equivalent circuits shown in FIGS. 3A, 3B, 4A and 4B.

[0028] FIGS. 3A and 3B illustrate the derivation of the Norton equivalent circuit of a nonlinear cell at port *j*. The load current I_{eq} and conductance G_{eq} are the Norton equivalent model of the neighboring linear network and are shown in FIG. 3A. Assuming the load current of port *j* is perturbed by an amount of ΔI_{eq} at the interface (FIG. 3B), the nodal voltage and branch current at port *j* will change accordingly due to this perturbation and can be derived from the solution of this nonlinear cell to provide the equivalent output conductance of the nonlinear cell,

$$\hat{G}_{eq} = \frac{I_j^{'} - I_j}{V_j^{'} - V_J} \qquad (3)$$

$v_j, v_j^{'}, I_j, I_j^{'}$

where are nodal voltages and branch currents at port *j* with different load currents I_{eq}, and $I_{eq}+\Delta I_{eq}$.

[0029] Similarly, the Norton equivalent model of the linear network at the interface can be constructed. FIGS. 4A and 4B illustrate the derivation of the Norton equivalent circuit of a linear network at the port j. Different from the solution

for the nonlinear device interface, it is not efficient to solve the linear network repeatedly for each interface port because the number of nodes in linear network is usually large. Instead, a discrete approximation method is used in which the Norton equivalent conductance G_{eq} and current I_{eq} of the linear network at the interface port are approximated from boundary conditions defined by the nodal voltages and branch currents of the corresponding nonlinear cell of previous iterations. Hence, the nonlinear cell nodal voltages V_1, V_2 and branch currents I_1, I_2 of two previous Newton-Raphson iterations have the following relation,

$$\begin{cases} \hat{V}_1 = \left(I_{eq} + \hat{I}_1\right)/G_{eq} \\ \hat{V}_2 = \left(I_{eq} + \hat{I}_2\right)/G_{eq} \end{cases} \quad (4)$$

The Norton equivalent circuit model G_{eq} and I_{eq} are then obtained by solving the above equation as below:

$$\begin{cases} G_{eq} = \dfrac{\hat{I}_2 - \hat{I}_1}{\hat{V}_2 - \hat{V}_1} \\ I_{eq} = G_{eq}\hat{V}_2 - \hat{I}_2 \end{cases} \quad (5)$$

[0030] FIG. 5 illustrates the flow of overall transient analysis based on the above simulation process. First, a circuit under analysis is represented by a computer-recognizable file and this circuit file is read by the computer in the "load circuit" step. Next at the "load device" step, the circuit file with all of its circuit devices is converted into a matrix form. At the "integration approximation" step, the differential circuit equations are approximated into sums of over time steps without time integrals. Each circuit device with a nonlinear behavior (e.g., a nonlinear I-V response) is linearized. After the integration approximation and Newton-Raphson linearization,

the linear networks and nonlinear components are solved separately. First, the linear networks are modeled as equivalent Norton circuits at the linear-nonlinear interface and nonlinear components are solved by the LU decomposition method with those boundary conditions. Next, all nonlinear components are modeled with respective Norton equivalent circuits and the linear networks are solved using the adaptive algebraic multigrid method. This linear-nonlinear iteration continues until the difference of interface port nodal voltages calculated from linear and nonlinear stages is smaller than the error tolerance at every interface port.

[0031] Hence, two iterations are performed in the process in FIG. 5. The first iteration is performed for the convergence at each linear-nonlinear interface which requires the boundary voltages calculated from two stages to be the same. The entire process iterates until the convergence is achieved. The second iteration is the iteration under the Newton-Raphson algorithm.

[0032] The convergence conditions for the two-stage Newton-Raphson algorithm can be derived as follows. Nodes in the circuit networks are classified into three different kinds of nodes. The internal nodes of linear networks are defined as linear nodes, the internal nodes of nonlinear cells are defined as non-linear nodes and the nodes connecting linear and nonlinear components are defined as interface nodes.

[0033] Based on the above classification of nodes, the linearized circuit equation according to the order of groups of the nodes can be expressed as follows:

$$\begin{bmatrix} G_1 & G_b & 0 \\ B_9 & B_1 + B_n & B_m \\ 0 & M_b & M \end{bmatrix} \begin{bmatrix} V_g \\ V_b \\ V_m \end{bmatrix} = \begin{bmatrix} b_g \\ b_b \\ b_m \end{bmatrix} \qquad (6)$$

where the subscripts g, b, and m refer to linear nodes, interface nodes, and nonlinear nodes, respectively.

Submatrices G_1, and M are Jacobian matrices corresponding to the linear nodes and nonlinear nodes. The submatrix B_1 denotes the Jacobian matrix contributed by linear circuits at the interface nodes. The submatrix B_n denotes the Jacobian matrix by nonlinear circuits at the interface nodes. Submatrices B_g, G_b, M_b and B_m represent the linearized relations between node groups. Vectors Vg, V_b, V_m are nodal voltages of linear nodes, interface nodes and nonlinear nodes, respectively. Vectors b_g, b_b, b_m denote the corresponding currents at linear nodes, interface nodes and nonlinear nodes, respectively.

[0034] FIG. 6A illustrates the equivalent circuit diagram for the above classification of nodes.

[0035] The two-stage Newton-Raphson approach is described by iterative equations,

$$\begin{cases} \begin{bmatrix} G_1 & G_b \\ B_g & B_1 + D_1 \end{bmatrix} \begin{bmatrix} V_g^{k+1} \\ V_b^{k+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} b_g \\ b_{b1} \end{bmatrix} \\ \begin{bmatrix} B_m + D_2 & B_m \\ M_b & M \end{bmatrix} \begin{bmatrix} V_b^{k+1} \\ V_m^{k+1} \end{bmatrix} = \begin{bmatrix} b_{b2} \\ b_m \end{bmatrix} \end{cases} k = 0,1,2\ldots \quad (7)$$

where

$$b_{b1}=b_b- B_mM^{-1}b_m-N_1V_b^{k},$$

$$b_{b2}=b_b-B_gG_1^{-1} b_g-N_2V_b^{k+1/2} ,$$

$$D_1 = \mathrm{Diag}(B_n-B_mM^{-1}M_b),$$

$$N_1 = B_n- B_mM^{-1} M_b - D_1,$$

$$D_2 = \mathrm{Diag}(B_1-B_gG_1^{-1} G_b) \text{ and}$$

$$N_2 = B_1 - B_gG_1^{-1}G_b - D_2.$$

FIGS. 6B and 6C further illustrate the equivalent circuit diagrams for equivalent circuits for the nonlinear nodes and linear nodes, respectively.

[0036] The convergence condition for the above iteration equations in Eq. (7) is as follows: the two-stage Newton-Raphson approach converges if

$$\| E_2^{-1} N_2 E_1^{-1} N_1 \| < 1$$

where

$$E_1 = B_n + D_2 - B_m M^{-1} M_b \text{ and}$$

$$E_2 = B_1 + D_1 - B_g G_1^{-1} G_b.$$

[0037] Referring to FIGS. 1 and 5, the equivalent linear network for the original circuit can be solved by a linear circuit simulation method. For example, various linear circuit simulations based on multigrids can be used. The basic idea of a multigrid method is to map the hard-to-damp low frequency error at fine level to easy-to-damp high frequency error at coarse level, solve the mapped problem at coarse level, and then map the error correction of coarse level back to fine level. A hierarchical grid structure with multiple levels is constructed to perform such multigrid computations. At each level, a forward iterative smoothing operator such as Gauss-Seidel erases high frequency errors. There are two kinds of multigrid methods: the geometric multigrid and the algebraic multigrid (AMG).

[0038] The network analysis methods described here are based on algebraic multigrid (AMG) methods described by W. L. Briggs in "A Multigrid Tutorial", SIAM 2000 and the Web site at http://www.llnl.gov/casc/people/henson/mgtut/ps/mgtut.pdf. The AMG is a multigrid method and is an efficient technique for solving partial differential equations. The geometric multigrid method generally requires regular mesh structures. AMG does not require a regular mesh structure and can apply to other non-regular structures. In at least this regard, the AMG is a good alternative to the geometric multigrid method. The coarsening and interpolation operations of the AMG are based on the matrix itself. This overhead may make the AMG less efficient than the geometric multigrid method if the problem analyzed has a regular mesh structure. One type of the AGM methods is the adaptive algebraic multigrid approach

as disclosed PCT Publication No. WO2004109452A3 for accurate analysis of the linear network at a high processing speed.

[0039] One implementation of the adaptive algebraic multigrid approach, for example, includes representing the linear circuit network by using two or more levels of grids with different numbers of nodes to represent the circuit network, applying a restriction mapping from one level to a next coarser level to propagate computation results of the one level to the next coarse level, and applying an interpolation mapping from one level to a next finer level to propagate computation results of the one level to the next finer level. In each level, an iterative smoothing operation is performed to obtain computation results of each level comprising states of nodes in each level. The above restriction mapping and the iterative smoothing operation from the finest level to the coarsest level and the interpolation mapping and the iterative smoothing operation from coarsest level back to the finest level are repeated for at least one time to obtain a solution to the circuit network.

[0040] Another implementation of the adaptive algebraic multigrid approach may include the following steps. An algebraic multigrid method is applied to a matrix representative of the linear circuit network to construct matrices with different degrees of coarsening grids. The regions in the circuit network with active circuit activities are represented by active grids and regions in the circuit network with less active circuit activities are represented by inactive grids. An iterative smoothing operation is then applied to an active grid more frequently than an inactive grid to reduce an amount of computation.

[0041] In yet another implementation of the adaptive algebraic multigrid approach, the linear circuit network is first represented by a matrix of nodes having fine nodes and coarse nodes. An adaptive coarse grid construction procedure is

applied to assign grid nodes in the matrix as either coarse grid nodes or fine grid nodes. This assignment is made according to (1) circuit activities and (2) a matrix structure of the matrix. Next, iterative smoothing operations are applied at selected local fine grids corresponding to active regions at a finest level obtained in the adaptive coarse grid construction procedure.

[0042] The adaptive algebraic multigrid approach can include the contribution of the inductance of the circuit networks in addition to the effect of the resistance of the network to provide an accurate analysis of the circuit behavior. This is because the effect of the circuit inductance can become comparable with the contribution of the resistance when the signal frequency increases beyond a certain level. The adaptive features in the grid coarsening and the error smoothing operations in the adaptive algebraic multigrid approach can significantly improve the processing speed.

[0043] In implementations, the above described techniques and their variations may be implemented as computer software instructions. Such software instructions may be stored in an article with one or more machine-readable storage media that are not connected to a computer, or stored in one or more machine-readable storage devices connected to one or more computers either directly or via a communication link. In operation, the instructions are executed by, e.g., one or more computer processors, to cause the machine to perform the described functions and operations for circuit analysis.

[0044] The above two-stage Newton-Raphson approach may be implemented as computer software instructions in C or the programming languages. Tests were conducted to illustrate performance of the two-stage Newton-Raphson approach using the C programming language. In the tests, the schemes of dynamic time step size control and integration approximation were based on the same schemes used in the Berkeley SPICE3f5. The

Berkeley BSIM3 was used for the transistor model in all test cases. The linear networks were solved by the adaptive multigrid method. Four industry designs were tested on a Linux machine with 2.6 GHz CPU and 4 Gigabytes memory.

[0045] A P/G network used in the tests contained an power ground network and a two-level H-tree clock. The circuit had 5400 inductors and 43700 mutual inductors. Coupling capacitors linked between power, ground and clock networks. FIG. 7A shows the waveform of one node on the clock. Transient simulation of 10 ns was completed in 1859 seconds, which is more than 20 times faster than SPICE3 as shown in Table I.

TABLE I

TRANSIENT SIMULATION RUNTIME

Examples	P/G Network	1K-cell	10K-cell	IO
#Nodes	29,100	10,200	123,600	1,062,000
#Transistors	720	6,500	69,000	56,600
Simulation Period	10ns	20ns	20ns	20ns
SPICE3(sec)	41323	2121	44293	N/A
Proposed Method(sec)	1859	261	3572	15337
Speedup	22.2	8.1	12.4	N/A

[0051] where transistor devices are dominant. The 1K cell circuit had 10,200 nodes and 6,500 transistors. The 10K cell circuit had 123,600 nodes and 69,000 transistors. It is assumed that an ideal power and ground supply is provided in those RC examples. The present technique took 261 seconds for 1K cell circuit and 3572 seconds for 10K cell circuit to finish 20-ns transient simulations. The processing speed was faster than that of SPICE3 by a factor of 8.1 and 12.4 for the 1K cell and 10K cell examples, respectively (Table I). The detailed transistor device BSIM3 model evaluation time became the bottleneck for these transistor devices dominant circuits. Further increase in the processing speed can be expected if a simplified device model is used.

[0052] In addition, the output waveforms were thoroughly tested against the Berkeley SPICE3. An accurate match with the waveform from the Berkeley SPICE3 was observed in both examples. FIG. 7B shows the transient waveform of one gate output in the 1K cell design.

[0053] In another test, a power/ground network with IO cells was simulated. This circuit had a large power/ground network with 1 million nodes and hundreds of big IO cells (6k transistors each). Berkeley SPICE3 and HSPICE fail to execute the simulation because of the excessive requirements on the memory size and computation time. The simulation software in C based on the present technique, however, successfully finished the transient analysis of 20ns in 4 hours. FIG. 8A shows input and output waveforms of one IO cell. FIG. 8B illustrates the voltage drop of a node on the power network.

[0054] Although only a few examples are described, other variations and enhancements can be made.

Claims

What is claimed are the techniques and systems as illustrated and described, including:

1. A method for analyzing a circuit network having transistors, comprising:

representing a circuit by an equivalent circuit that comprises at least one linear network and at least one nonlinear cell to interface with each other;

solving circuit equations for the at least one linear network and the at least one nonlinear cell, respectively; and

matching solutions to the at least one linear network and the at least one nonlinear cell at an interface between the at least one linear network and the at least one nonlinear cell.

2. The method as in claim 1, wherein solving the at least one nonlinear cell comprises:

representing the at least one linear network by a first Norton equivalent circuit that is connected to the at least one nonlinear cell; and

applying a LU decomposition to the at least one nonlinear cell under a boundary condition with the first Norton equivalent circuit to find a solution to the at least one nonlinear cell.

3. The method as in claim 1, wherein solving the at least one linear network comprises:

representing the at least one nonlinear cell by a second Norton equivalent circuit that is connected to the at least one linear network; and

applying a linear circuit solver to the at least one linear network under a boundary condition with the second Norton equivalent circuit to find a solution to the at least one linear network.

4. The method as in claim 3, wherein the linear circuit solver is an adaptive algebraic multigrid linear solver.

5. The method as in claim 4, wherein the multigrid linear solver uses a geometric multigrid.

6. The method as in claim 4, wherein the multigrid linear solver uses an algebraic multigrid.

7. The method as in claim 6, wherein the multigrid linear solver is to

represent the at least one linear network by using a matrix of nodes having fine nodes and coarse nodes;

apply an adaptive coarse grid construction procedure to assign grid nodes in the matrix as either coarse grid nodes or fine grid nodes according to (1) circuit activities and (2) a matrix structure of the matrix to construct a plurality of levels of grids with different numbers of nodes to respectively represent the linear network; and

apply iterative smoothing operations at selected local fine grids corresponding to active regions at a finest level obtained in the adaptive coarse grid construction procedure.

8. The method as in claim 6, wherein the multigrid linear solver is to

apply an algebraic multigrid method to a matrix representative of the at least one network to construct a plurality of matrices with different degrees of coarsening grids;

represent regions in the linear network exhibiting active circuit activities with active grids and regions in the linear network exhibiting less active circuit activities with inactive grids; and

perform an iterative smoothing operation in an active grid more frequently than in an inactive grid to reduce an amount of computation.

9. The method as in claim 1, further comprising:

imposing conditions on voltages and currents in one or more internal circuit nodes in the at least one linear network, one or more nonlinear nodes in the at least one nonlinear cell, and one or more interface nodes at an interface between the at least one linear network and the at least one nonlinear cell to ensure convergence of the solutions to the at least one linear network and the at least one nonlinear cell.

10. A method for analyzing a circuit network having transistors, comprising:

representing a circuit by an equivalent circuit that comprises at least one linear network and at least one nonlinear cell to interface with each other;

using a Norton equivalent circuit to represent an interface between the at least one linear network and the at least one nonlinear cell;

applying a Newton-Raphson iteration to solve circuit equations for the at least one linear network and the at least one nonlinear cell; and

matching solutions to the at least one linear network and the at least one nonlinear cell at the interface between the at least one linear network and the at least one nonlinear cell.

11. The method as in claim 10, wherein solving the at least one nonlinear cell comprises:

applying a LU decomposition to the at least one nonlinear cell.

12. The method as in claim 10, wherein solving the at least one linear network comprises:

applying a linear circuit solver to the at least one linear network.

13. The method as in claim 12, wherein the linear circuit solver is a multigrid linear solver.

14. The method as in claim 13, wherein the multigrid linear solver uses a geometric multigrid.

15. The method as in claim 13, wherein the multigrid linear solver uses an algebraic multigrid.

16. An article comprising at least one machine-readable medium that stores machine-executable instructions, the instructions causing a machine to:

represent a circuit by an equivalent circuit that comprises at least one linear network and at least one nonlinear cell to interface with each other;

solve circuit equations for the at least one linear network and the at least one nonlinear cell, respectively; and

match solutions to the at least one linear network and the at least one nonlinear cell at an interface between the at least one linear network and the at least one nonlinear cell.

17. The article as in claim 16, wherein the instructions further cause the machine to:

represent the at least one linear network by a first Norton equivalent circuit that is connected to the at least one nonlinear cell; and

apply a LU decomposition to the at least one nonlinear cell under a boundary condition with the first Norton

equivalent circuit to find a solution to the at least one nonlinear cell.

18. The article as in claim 16, wherein the instructions further cause the machine to:

represent the at least one nonlinear cell by a second Norton equivalent circuit that is connected to the at least one linear network; and

apply a linear circuit solver to the at least one linear network under a boundary condition with the second Norton equivalent circuit to find a solution to the at least one linear network.

19. The article as in claim 16, wherein the instructions further cause the machine to apply a multigrid linear solver to solve for the at least one linear network.

20. The article as in claim 19, wherein the instructions further causing the machine to:

apply an algebraic multigrid method to a matrix representative of the at least one linear network to construct a plurality of matrices with different degrees of coarsening grids;

divide the at least one linear network into active regions and inactive regions according to circuit activities; and

perform an iterative smoothing operation in an active region more frequently than in an inactive region.

FIG. 1

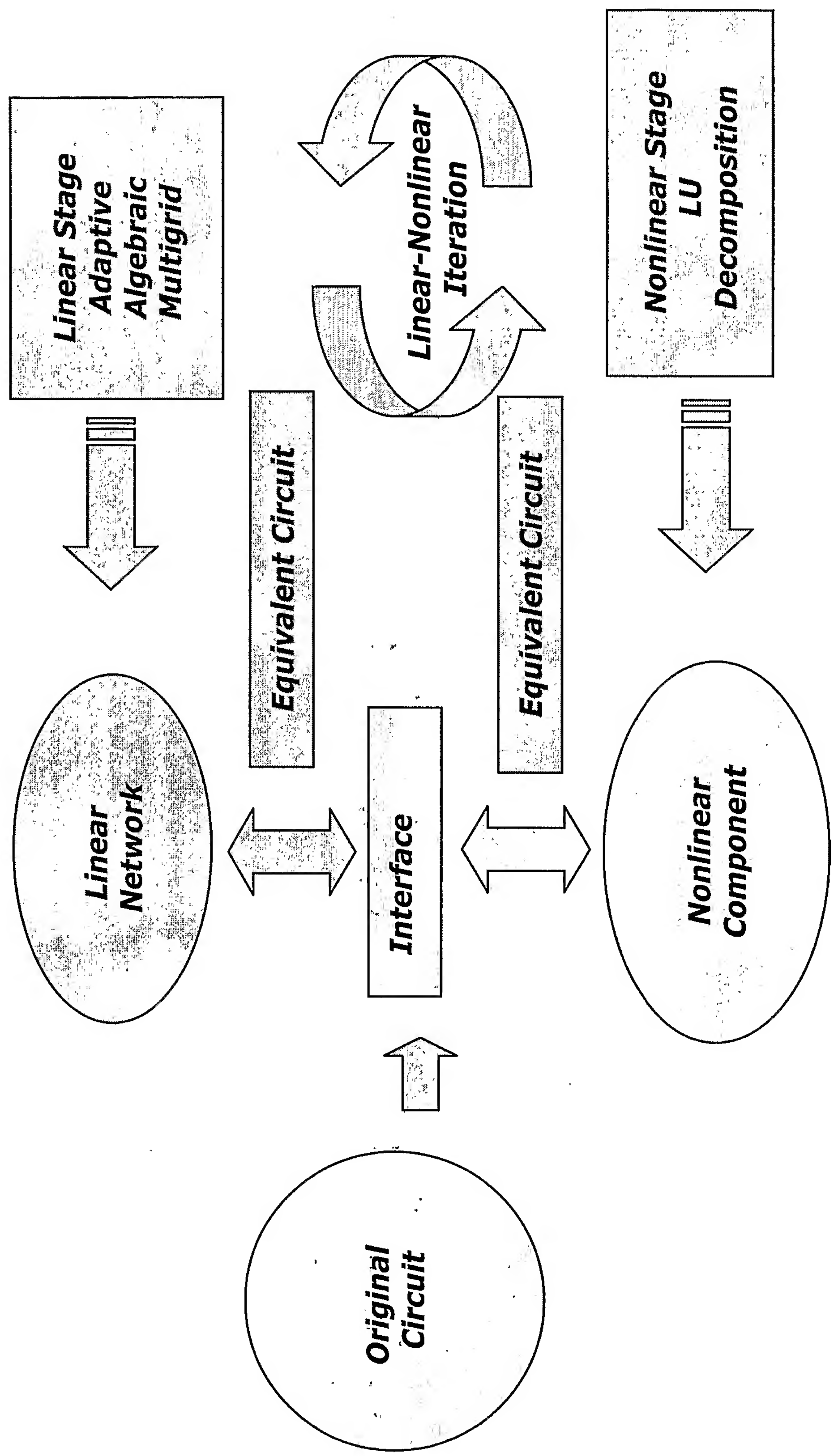

Original Circuit
Interface
Linear Network
Nonlinear Component
Equivalent Circuit
Equivalent Circuit
Linear Stage Adaptive Algebraic Multigrid
Linear-Nonlinear Iteration
Nonlinear Stage LU Decomposition

FIG. 1C

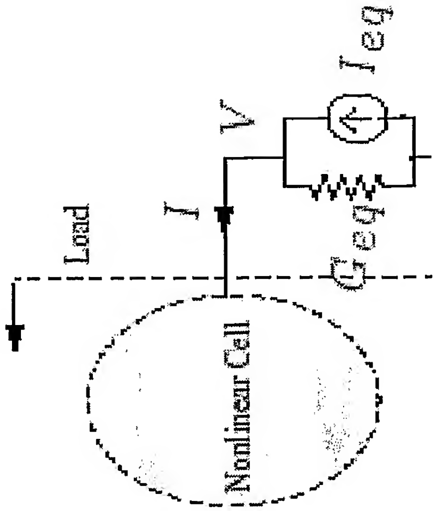


FIG. 1B

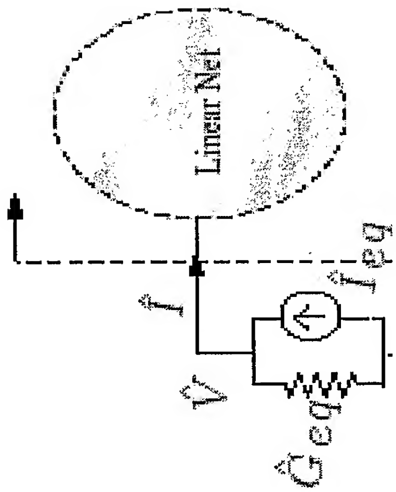


FIG. 1A

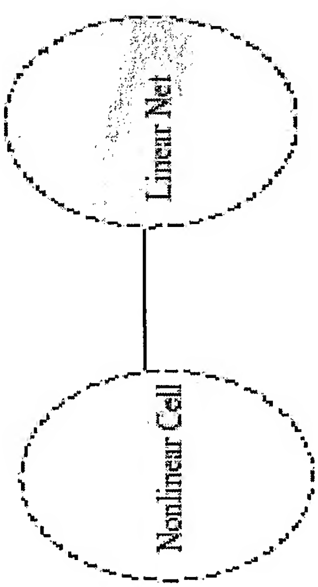


FIG. 1E

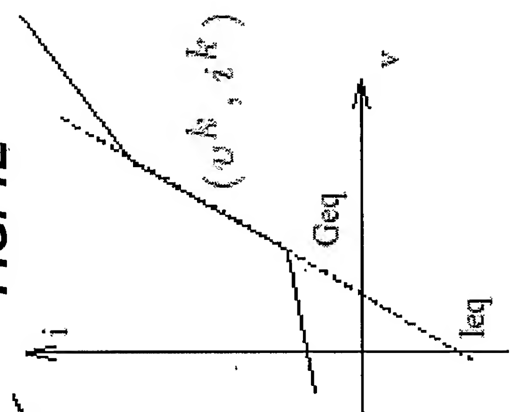


FIG. 1D

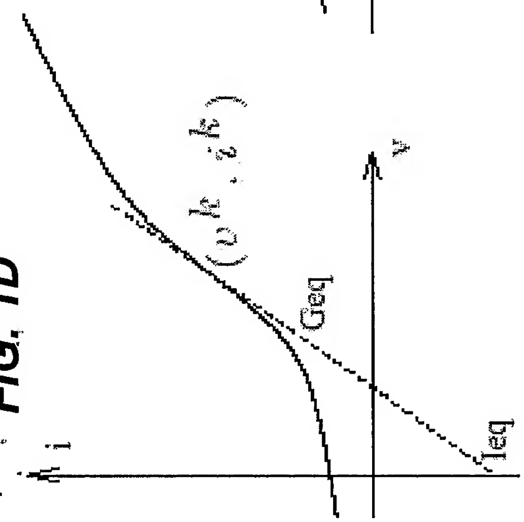
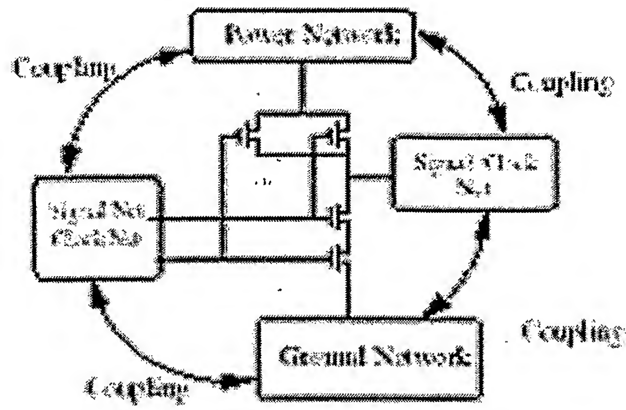


FIG. 2A



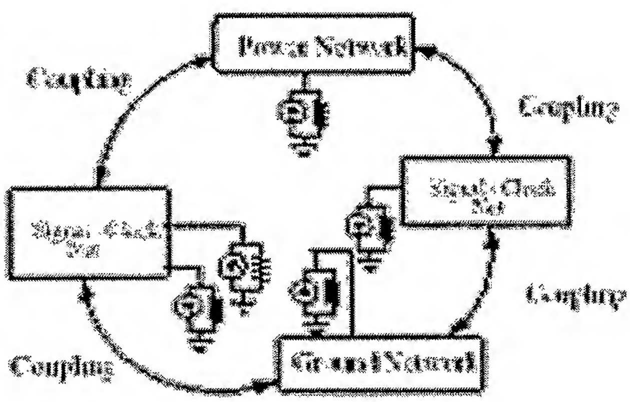


FIG. 2B

FIG. 2C

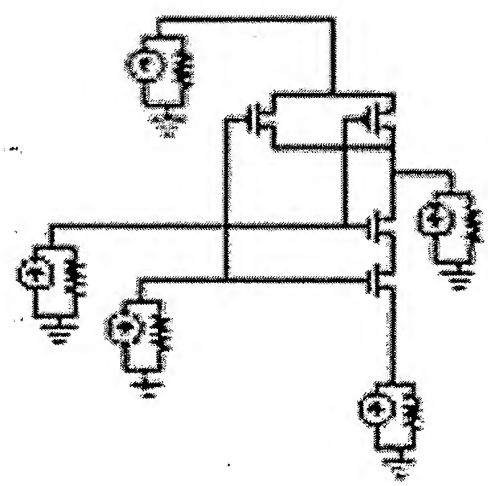
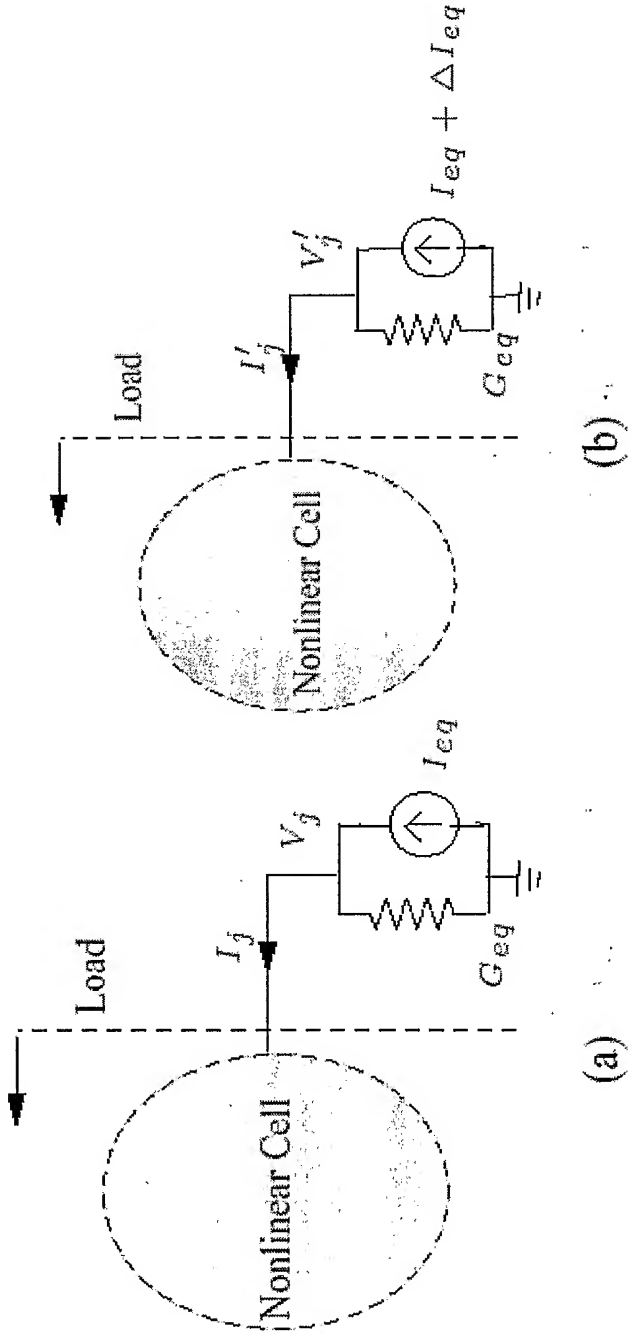


FIG. 3A

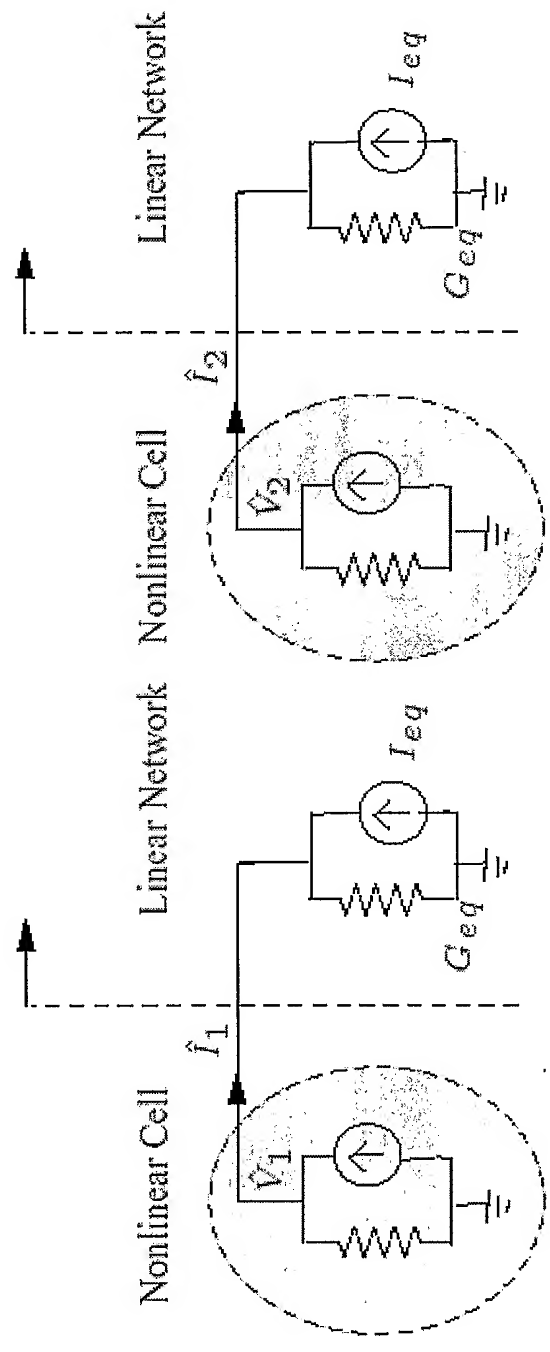
FIG. 3B



$$\hat{G}_{eq} = \frac{I'_j - I_j}{V'_j - V_j}$$

FIG. 4A

FIG. 4B



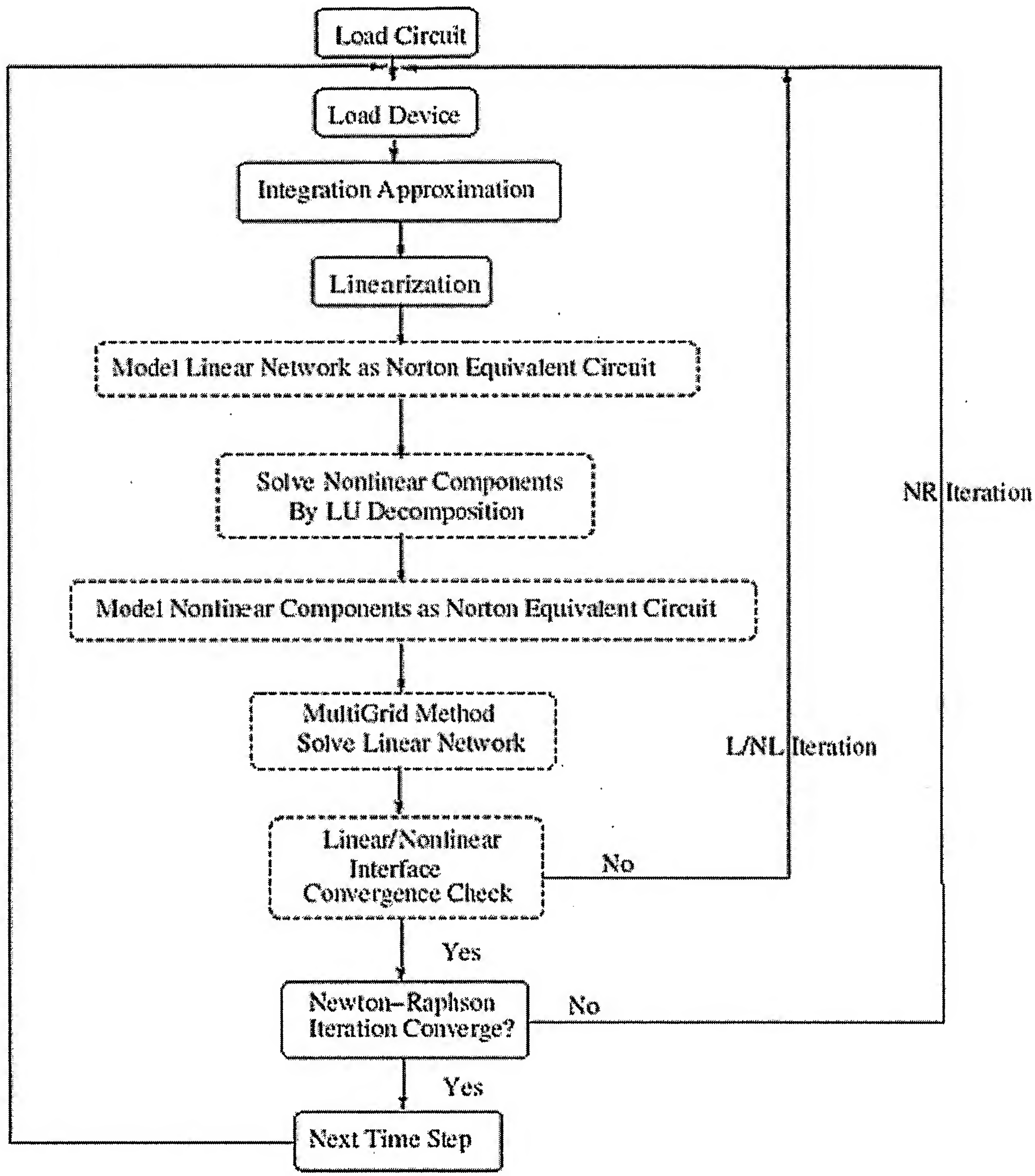
(a)

(b)

$$\begin{cases} \hat{V}_1 = (I_{eq} + \hat{I}_1)/G_{eq} \\ \hat{V}_2 = (I_{eq} + \hat{I}_2)/G_{eq} \end{cases}$$

$$\begin{cases} G_{eq} = \dfrac{\hat{I}_2 - \hat{I}_1}{\hat{V}_2 - \hat{V}_1} \\ I_{eq} = G_{eq}\hat{V}_2 - \hat{I}_2 \end{cases}$$

FIG. 5



Load Circuit
Load Device
Integration Approximation
Linearization
Model Linear Network as Norton Equivalent Circuit
Solve Nonlinear Components
By LU Decomposition
Model Nonlinear Components as Norton Equivalent Circuit
MultiGrid Method
Solve Linear Network
Linear/Nonlinear
Interface
Convergence Check
No
L/NL Iteration
Yes
Newton–Raphson
Iteration Converge?
No
NR Iteration
Yes
Next Time Step

FIG. 6A

$$\begin{bmatrix} G_l & G_b & 0 \\ B_g & B_l + B_n & B_m \\ 0 & M_b & M \end{bmatrix} \begin{bmatrix} V_g \\ V_b \\ V_m \end{bmatrix} = \begin{bmatrix} b_g \\ b_b \\ b_m \end{bmatrix}$$

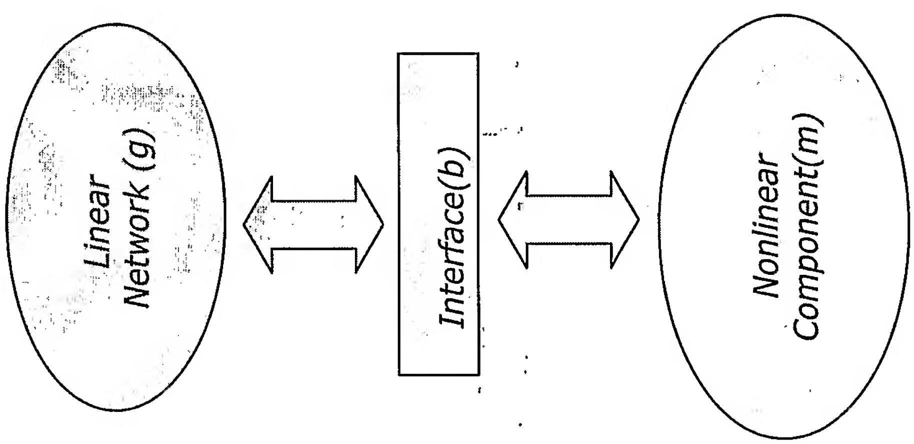


FIG. 6B

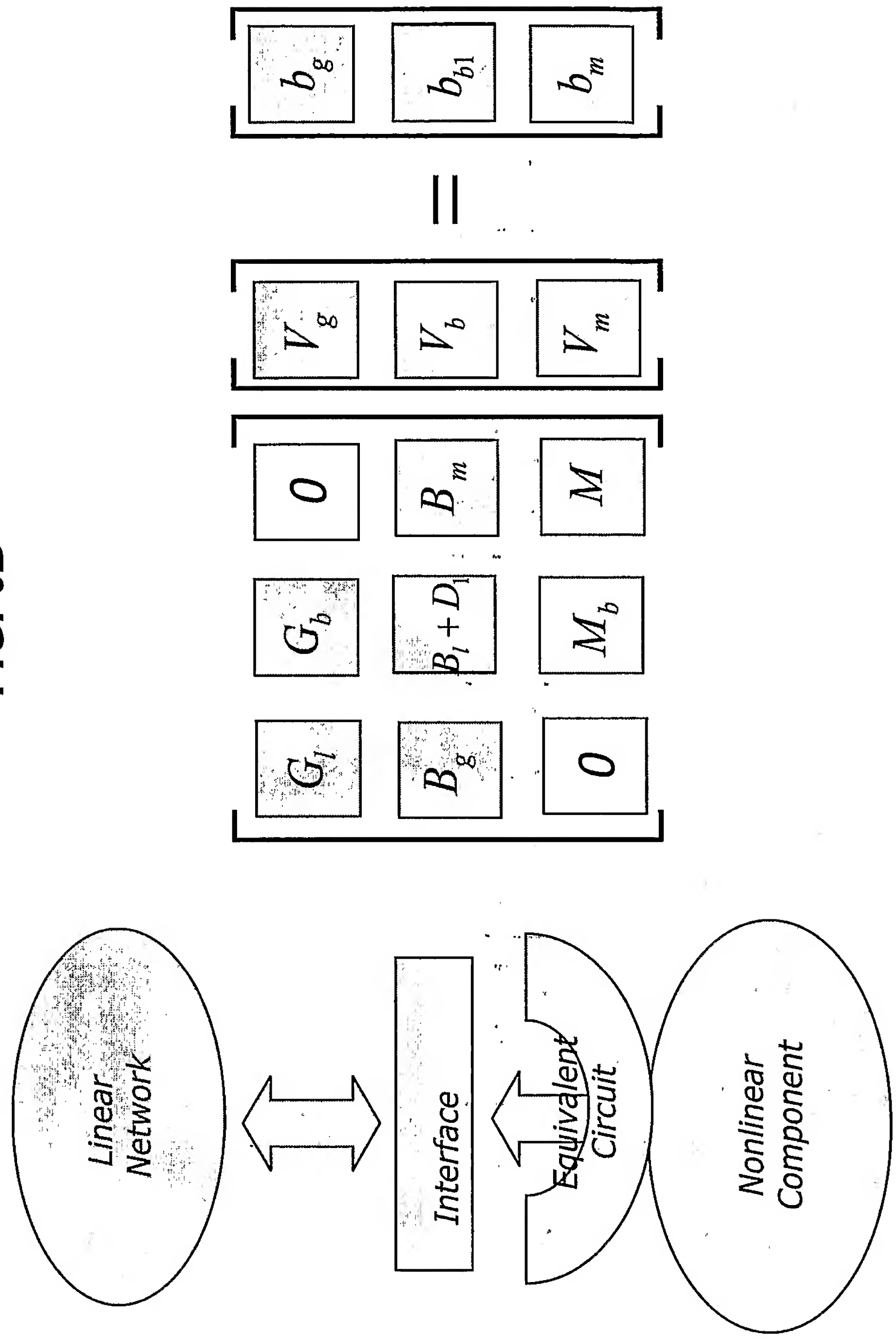

$\begin{bmatrix} G_l & G_b & 0 \\ B_g & B_l + D_l & B_m \\ 0 & M_b & M \end{bmatrix} \begin{bmatrix} V_g \\ V_b \\ V_m \end{bmatrix} = \begin{bmatrix} b_g \\ b_{b1} \\ b_m \end{bmatrix}$
Linear Network
Interface
Equivalent Circuit
Nonlinear Component

FIG. 6C

$$\begin{bmatrix} G_l & G_b & 0 \\ B_g & D_2+B_n & B_m \\ 0 & M_b & M \end{bmatrix} \begin{bmatrix} V_g \\ V_b \\ V_m \end{bmatrix} = \begin{bmatrix} b_g \\ b_{b2} \\ b_m \end{bmatrix}$$

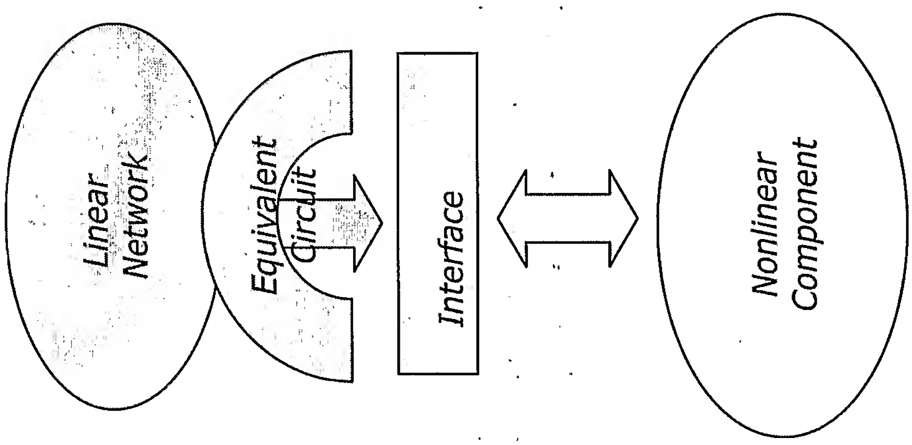


FIG. 7A

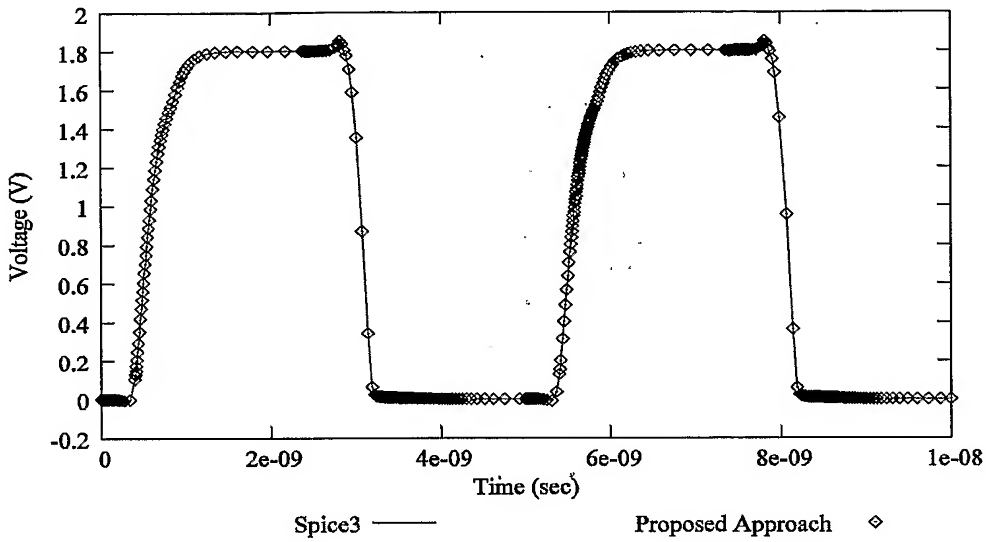

2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Voltage (V)
0
2e-09
4e-09
6e-09
8e-09
1e-08
Time (sec)
Spice3
Proposed Approach

FIG. 7B

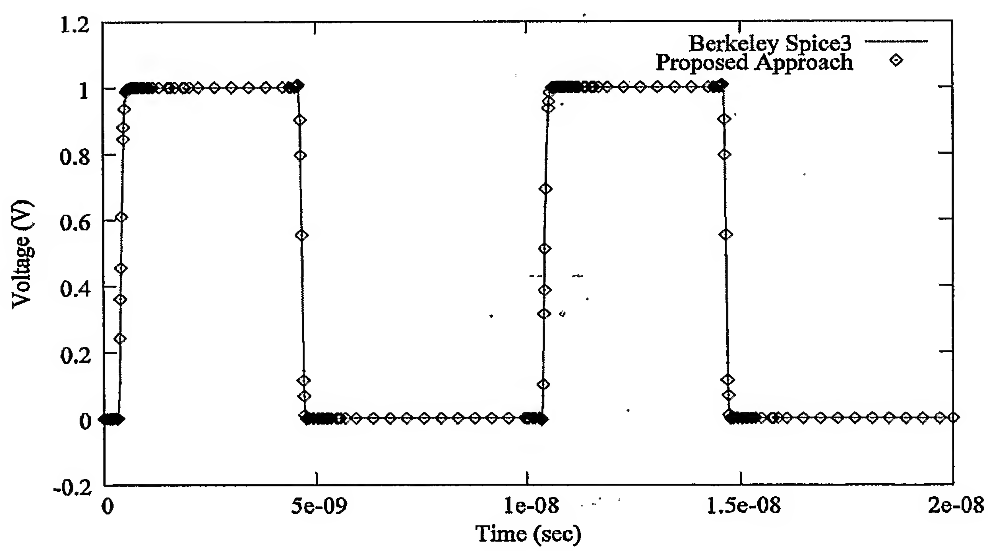

1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Voltage (V)
Berkeley Spice3
Proposed Approach
0
5e-09
1e-08
1.5e-08
2e-08
Time (sec)

FIG. 8A

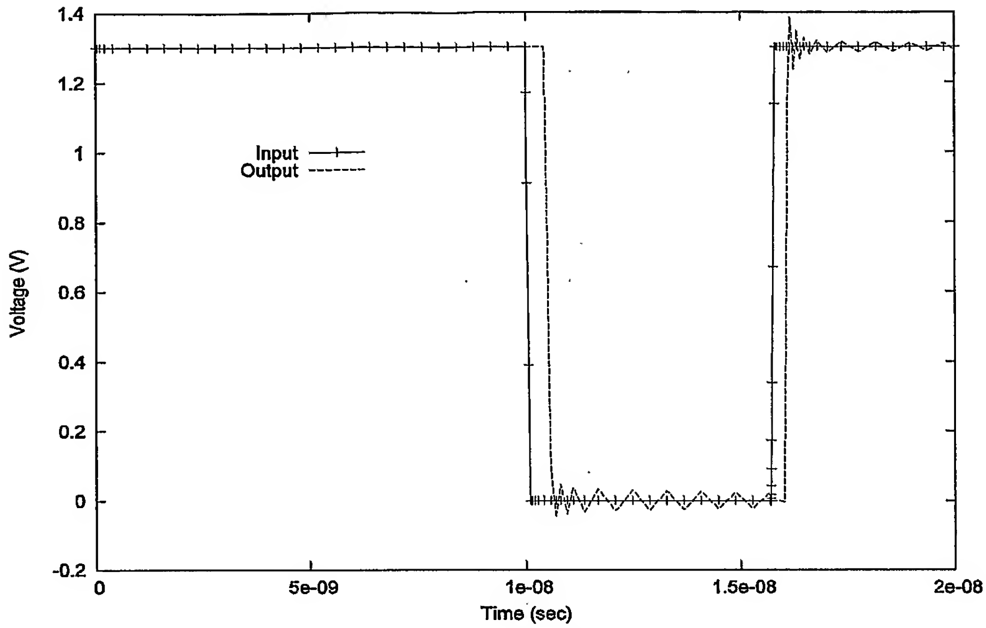
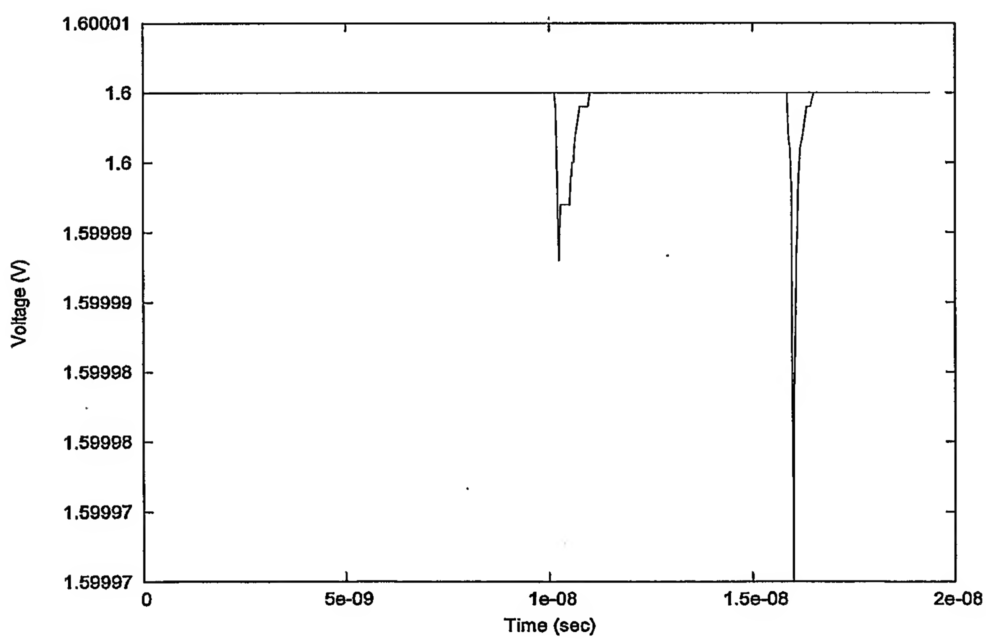

1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
Voltage (V)
Input
Output
0
5e-09
1e-08
1.5e-08
2e-08
Time (sec)

FIG. 8B


1.60001
1.6
1.6
1.59999
1.59999
1.59998
1.59998
1.59997
1.59997
Voltage (V)
0
5e-09
1e-08
1.5e-08
2e-08
Time (sec)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US05/20369

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 17/50 (2006.01
USPC - 703/14
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 703/3,11,12,13,14, 102; 716/1,11,17; 708/6,13,443,602,801,802,804

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
PUBWEST--PGPB, USPT, EPAB and JPAB--searched linear, non-linear, network, circuit, multigrid, VCSI, Newton-Raphson, transistors, circuit equations, non-linear cell, integrated circuit

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X ——— Y	US 2004/0044510 A1 (ZOLOTOV et al.) 4 March 2004 (04.03.2004), see whole document	1, 9, 16 and 19-20 ——— 2-8, 10-15, 17 and 18
Y	US 6,662,149 B1 (DEVGAN et al.) 9 December 2003 (09.12.2003), see Col. 3 Line 35-47	2-8, 10-15, 17 and 18
A	US 5,694,052 A (SAWAI et al.) 2 December 1997 (02.12.1997), see entire document	1-20
A	US 6,665,849 B2 (MEURIS et al.) 16 December 2003 (16.12.2003), see entire document	1-20
A	US 6,141,676 A (RAMIREZ-ANGULO et al.) 31 October 2000 (31.10.2000), see entire document	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
	02 JUN 2006

Name and mailing address of the ISA/US	Authorized officer:
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Blaine R. Copenheaver Telephone No. 571-272-7774